

CLAIMS

What is claimed is:

1. A method for loading methods into a virtual machine, the methods contained in one or more classes, the method including:
 - recursively scanning through application code beginning at an application entrypoint to determine which methods may be called and the classes to which they correspond;
 - storing identifiers corresponding to said methods which may be called in a method usage map organized by classes;
 - consulting said method usage map upon execution of the virtual machine; and
 - selectively loading only those methods contained in said method usage map into memory in the virtual machine.
2. The method of claim 1, wherein said recursively scanning includes:
 - finding all methods referenced in said application code; and
 - finding all methods referenced in methods referenced in said application code.
3. The method of claim 1, wherein said storing includes storing said method usage map in a file located in secondary storage.
4. The method of claim 3, wherein each of the classes is stored in secondary storage.

5. The method of claim 1, wherein said recursively scanning includes statically determining which methods of the classes may actually be used by the application.
6. The method of claim 1, wherein said recursively scanning includes handling method polymorphism.
7. A method for generating a method usage map for use in loading methods into a virtual machine, the methods contained in one or more classes, the method including:
 - recursively scanning through application code beginning at an application entrypoint to determine which methods may be called and the classes to which they correspond; and
 - storing identifiers corresponding to said methods which may be called in a method usage map organized by classes.
8. The method of claim 7, wherein said recursively scanning includes:
 - finding all methods referenced in said application code; and
 - finding all methods referenced in methods referenced in said application code.
9. The method of claim 7, wherein said storing includes storing said method usage map in a file located in a database.
10. The method of claim 9, wherein each of the classes is stored in the database.

11. The method of claim 7, wherein said recursively scanning includes statically determining which methods of the classes may actually be used by the application.
12. The method of claim 7, wherein said recursively scanning includes handling method polymorphism.
13. A method for loading methods into a virtual machine, the methods contained in one or more classes, the method including:
- consulting said method usage map stored in a database upon execution of the virtual machine; and
 - selectively loading identifiers corresponding to only those methods contained in said method usage map into memory in the virtual machine.
14. An apparatus for loading methods into a virtual machine, the methods contained in one or more classes, the apparatus including:
- an application code method recursive scanner;
 - a method usage map generator coupled to said application code method recursive scanner;
 - a secondary storage;
 - a method usage map storer coupled to said method usage map generator and to said secondary storage;
 - a method usage map consuler coupled to said secondary storage; and
 - a virtual machine selective method loader coupled to the method usage map consuler.

15. An apparatus for generating a method usage map for use in loading methods into a virtual machine, the methods contained in one or more classes, the apparatus including:

an application code method recursive scanner;

a method usage map generator coupled to said application code method recursive scanner;

a secondary storage; and

a method usage map storer coupled to said method usage map generator and to said secondary storage.

16. An apparatus for loading methods into a virtual machine, the methods contained in one or more classes, the apparatus including:

a database;

a method usage map consuler coupled to said database; and

a virtual machine selective loader coupled to the method usage map consuler.

17. An apparatus for loading methods into a virtual machine, the methods contained in one or more classes, the apparatus including:

means for recursively scanning through application code beginning at an application entryptoint to determine which methods may be called and the classes to which they correspond;

means for storing identifiers corresponding to said methods which may be called in a method usage map organized by classes;

means for consulting said method usage map upon execution of the virtual machine; and

means for selectively loading only those methods contained in said method usage map into memory in the virtual machine.

18. The apparatus of claim 17, wherein said means for recursively scanning includes:
means for finding all methods referenced in said application code; and
means for finding all methods referenced in methods referenced in said application code.
19. The apparatus of claim 17, wherein said means for storing includes means for storing said method usage map in a file.
20. The apparatus of claim 19, wherein each of the classes is stored in a secondary storage.
21. The apparatus of claim 17, wherein said means for recursively scanning includes means for statically determining which methods of the classes may actually be used by the application.
22. The apparatus of claim 17, wherein said means for recursively scanning includes means for handling method polymorphism.
23. An apparatus for generating a method usage map for use in loading methods into a virtual machine, the methods contained in one or more classes, the apparatus including:
means for recursively scanning through application code beginning at an application entrypoint to determine which methods may be called and the classes to which they correspond;
and

means for storing identifiers corresponding to said methods which may be called in a method usage map organized by classes.

24. The apparatus of claim 23, wherein said means for recursively scanning includes:
means for finding all methods referenced in said application code; and
means for finding all methods referenced in methods referenced in said application code.
25. The apparatus of claim 23, wherein said means for storing includes means for storing said method usage map in a file located in a database.
26. The apparatus of claim 25, wherein each of the classes is stored in the database.
27. The apparatus of claim 23, wherein said means for recursively scanning includes means for statically determining which methods of the classes may actually be used by the application.
28. The apparatus of claim 23, wherein said means for recursively scanning includes handling method polymorphism.
29. An apparatus for loading methods into a virtual machine, the methods contained in one or more classes, the apparatus including:
means for consulting said method usage map stored in a database upon execution of the virtual machine; and

means for selectively loading only those methods contained in said method usage map into memory in the virtual machine.

30. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for loading methods into a virtual machine, the methods contained in one or more classes, the method including:

recursively scanning through application code beginning at an application entrypoint to determine which methods may be called and the classes to which they correspond;

storing identifiers corresponding to said methods which may be called in a method usage map organized by classes;

consulting said method usage map upon execution of the virtual machine; and

selectively loading only those methods contained in said method usage map into memory in the virtual machine.

31. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method generating a method usage map for use in loading methods into a virtual machine, the methods contained in one or more classes, the method including:

recursively scanning through application code beginning at an application entrypoint to determine which methods may be called and the classes to which they correspond; and

storing identifiers corresponding to said methods which may be called in a method usage map organized by classes.

32. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method for loading methods into a virtual machine, the methods contained in one or more classes, the method including:

consulting said method usage map stored in a database upon execution of the virtual machine; and

selectively loading only those methods contained in said method usage map into memory in the virtual machine.